**Marking Scheme:**

Ensure that you follow the processes and guidelines taught in class in order to produce high quality work. Do not just focus on getting the program working. This assessment rubric provides you with the characteristics of exemplary, competent, marginal and unacceptable work in relation to task criteria.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Exemplary** | **Competent** | **Marginal** | **Unacceptable** |
| **Technical writing** The textbook as an example of technical writing. | **3:** Document is correctly formatted. Technical writing is clear and appropriate. Correct amount of writing for each section. | **2:** Minor errors in formatting. Technical writing is mostly correct and appropriate. Mostly correct amount or writing for each section. | **1:** Many errors in layout and/or formatting. Technical writing is mostly incorrect. | **0:** Too many errors in formatting. Technical writing is mostly incorrect, and/or mostly incorrect amount or writing for each section. |
| **Analysis** | **3:** All user stories are correct, including formatting and wording | **2:** User stories are mostly correct, including formatting and wording | **1**: Many missing or incorrect user stories. | **0**: Too many missing or incorrect user stories. |
| **Testing** | **15:** Software solution works correctly as per the supplied specification. All user stories are tested. | **10:** Solution mostly works correctly as per the supplied specification. Few missing tests. | **5:** Solution partially satisfies the supplied specification. Many missing tests. | **0**: Software solution mostly does not satisfy the supplied specification. |
| **GitHub (Configuration management)** | **3:** Five or more commits per week with meaningful code changes, and commit comments. | **2:** Two-four commits per week with mostly meaningful code changes, and commit comments. | **1:** One or less commits per week with few meaningful code changes, and commit comments. | **0:** Missing, or only 1-3 commits. |
| **Solution quality**  **(Beyond specification)** | **3:** User friendly, clear and meaningful inputs and outputs. Error checking of inputs, error reporting. | **2:** Mostly clear and meaningful inputs and outputs. Some error checking of input, error reporting. | **1:** Mostly unclear and/and not meaningful inputs and outputs. Minimal error checking of inputs and/or error reporting. | **0:** No error checking of inputs, no error reporting. |
| **Code readability** | **3:** Classes, methods, variables are named clearly to support code readability.  Code contains inline comments where  helpful. 0-2 above errors in ten randomly inspected files. | **2:**  3-5 code readability errors in ten randomly inspected files. | **1**: 6-8 code readability errors in ten randomly inspected files. | **0:** more than 8 code readability errors in ten randomly inspected files. |
| **Coding standards** | **3:** 0-2 java style errors (including formating) in ten randomly inspected files. | **2:** 3-5 java style errors in ten randomly inspected files. | **1:** 6-8 java style errors in ten randomly inspected files. | **0:** more than 8 java style errors in ten randomly inspected files. |
| **Code quality** | **3:** 0-2 violations of textbook recommendations (e.g. DO/DONOTs), up to week-5 of lectures. | **2:** 3-5 violations of textbook recommendations. | **1**: 6-8 violations of textbook recommendations. | **0**: more than 8 violations of textbook recommendations. |